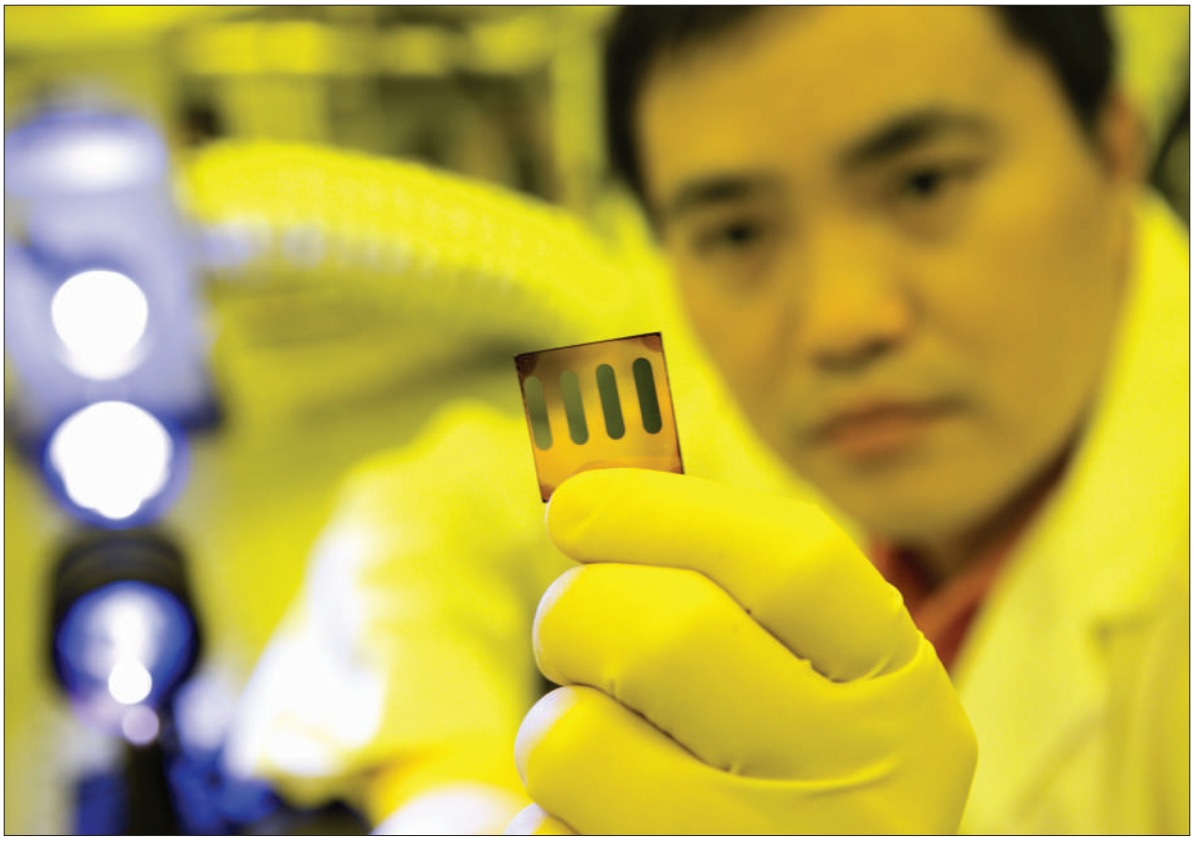

Singapore Journal of

Scientific Research

ISSN: 2010-006x

science alert

http://scialert.net/sjsr

OPEN ACCESS Singapore Journal of Scientific Research

ISSN 2010-006x
DOI: 10.3923/sjsres.2019.45.51

Research Article
Sentiment Analysis on Email Database Corpus-based Approach

[1]Mounika Kandukuri and [2]V.V. Hara Gopal

[1]Department of Statistics, University College of Science, Osmania University, Hyderabad, India
[2]Department of Mathematics, Birla Institute of Technology and Science, Pilani, Hyderabad Campus, Hyderabad, India

Abstract

Background and Objective: Sentiment analysis for social media sites and web data has been an exploding area in text mining but the research in the area of email sentiment analysis is not to that extent though it is extensively used in communication in our day-to-day tasks. The objective of this study was to perform sentiment analysis on email data to identify emotional intents expressed in emails. **Materials and Methods:** This study proposes a framework for email sentiment analysis using document-term matrix and the sentiment classification is conducted through lexicon-based approach using tidy text package. **Results:** The results are indicative to positive and negative keywords that contribute to each sentiment in email data. **Conclusion:** The proposed framework helps in identifying the different viewpoints of people expressed in emails which further helps organizations for better decision approach to meet customer expectations.

Key words: Email sentiment analysis, opinion mining, corpus, document-term matrix, tidy text package, lexicon approach

Citation: Mounika Kandukuri and V.V. Hara Gopal, 2019. Sentiment analysis on email database corpus-based approach. Singapore J. Sci. Res., 9: 45-51.

Corresponding Author: V.V. Hara Gopal, Department of Mathematics, Birla Institute of Technology and Science, Pilani, Hyderabad Campus, Hyderabad, India

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

From the past two decades right from the evolution of World Wide Web social media sites playing a vital role in every organization which obtaining large amounts of data in the form of editorials, emails, videos, tweets, voice recordings, books, Web pages and digital libraries. People living around started expressing their perspectives, viewpoints and thoughts through these online blogs hence, with the increase in the usage of the internet there is a tremendous growth in the volume of the data on the web, blogging and microblogging sites are emerging as an origin of a different kind of information[1]. This explosive growth of digital messages on social media can offer a wealth of information but data obtained is irregular, uninterruptible and ambiguous, thus making it difficult to analyze using traditional computing means for extracting hidden information and for making decisions[2].

Peoples also more enthusiastically started to share facts of their experiences, lives, experiences and thoughts with the entire world, the views shared by them can be sometimes positive attitude and sometimes in a negative attitude. Those views are renowned as sentiments, hence, detecting the sentiment from the views of people is termed as Sentiment analysis, it is also known as opinion mining (opinion mining-it is a study strives to identify the beliefs and perceptions of reality using computational methods from the text)[3]. This process contains the concepts of natural language processing (NLP) and information theory, also includes machine learning, which can be done at 3 levels like document level, sentence level, aspect (feature level)[4]. Document-level works well if opinion to be expressed about a single entity as positive or negative. Sentence level deals better when subjective (emotions, views or feelings) and objective (factual information) categorization of sentences is required finally aspect (feature) level is done when dealing with different features of products and targets on, to know what exactly people like or dislike the various features of object[4,5]. From past decades due to the development of machine learning methods in NLP and development of techniques and algorithms for sentiment analysis the applications of sentiment analysis has been extended to various fields[6,7].

In general sentiment analysis problem is composed of 2 major steps: Feature selection and sentiment classification. Features such as terms and their related frequency are extracted through document term matrix, term frequency-inverse document frequency and bag of words (BoWs). Hence, feature extraction algorithms include term frequency-inverse document frequency (TF-IDF), document-term matrix (DTM) and bag of words model (Bows)[6,7] and sentiment classification is composed of machine learning approaches, lexicon-based approaches, hybrid based approaches[6,7].

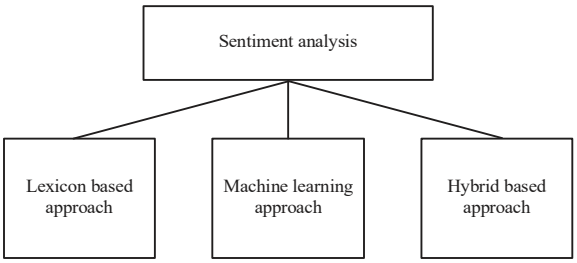

Fig. 1: Classification of sentiment analysis techniques

While experimental results demonstrate that the document term matrix with tidy text package and lexicon-based approach for sentiment classification performs well for email analysis. Hence, in this paper email sentiment analysis a little attempt has been done designing a framework using document term matrix as feature selection method and considering lexicon-based approach as classification approach to identify the emotional intent words from emails as shown in Fig 1.

The objective of this study to perform sentiment analysis on email data to identify emotional intents expressed in emails.

MATERIALS AND METHODS

This section of email sentiment analysis includes several techniques including Data and its preprocessing, feature selection, sentiment classification. The detailed content of techniques is explained below.

Data collection: The data used for this study is RS Agro Company, Hyderabad, October, 2016 Email database where customers quoted of their views about products and their services.

Sentiment analysis of the above email database can be followed up easily using a corpus-based approach but since all emails collected are dumped in one text file including their respective guide pattern and time stamp pattern. Using their respective unique identifiers different customer emails are brought down into text files where further analysis can be done by incorporating Corpus of brought down text files. At present scenario, As R is well known offering a wide range of statistical methods with sufficient computing environments; we use R software.

Appendix 1

```
con<-file("filename", "rb")
rl<-readLines(con, encoding = "UTF-16", skipNul = TRUE)
listOfemails<-1:as.integer(length(rl)/2)
numberOfemails<-(listOfemails-1)*2+1
issueemails<-vector(mode = "character", length = as.integer(length(rl)/2))
countIssueemails<-1
for(recCount in 1:length(rl))
{
if(nchar(rl[recCount]) > 0)
{
issueemails[countIssueemails]<-rl[recCount]
countIssueemails<-countIssueemails+ 1
}
}
rm(rl)
close.connection(con)
```

As each individual email is with guid pattern and time stamp pattern the pattern should be in format as

```
guidPattern<-"[0-9a-z]{8}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{4}-[0-9a-z]{12}"
```

Example : 4b6f0304-4941-4835-aed8-9f8c32d3ff00

```
timestampPattern<-"[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}"
```

Example : 2013-10-17 12:01:33

Now using "regexpr" function present in R environment is used which identifies whether the specified pattern in the line is within a character vector, searching each element separately,

```
pOfGuid<-regexpr(guidPattern, issueemails)
positionOfGuid<-as.vector(1:length(listOfemails))
```

The above Function results in number of emails in file based on guid and using code below we will create document ids based on guid.

```
for(recCount in 1:length(issueRecs)) positionOfGuid[recCount]<-pOfGuid [recCount][1]

GUID<-vector(mode = "character", length=length(positionOfGuid))
docId<-vector(mode = "integer", length=length(positionOfGuid))

for(recCount in 1:length(issueemails))
{
GUID[recCount]<substring(issueemails[recCount],positionOfGuid[recCount],p
ositionOfGuid[recCount]+35)
docId[recCount]<-recCount
}
```

Same "regexpr" function is used for timestamp pattern to extract based on date and time.

```
pOfTimeStamp<-regexpr(timeStampPattern, issueemails)
positionOfTimeStamp<-as.vector(1:length(listOfemails)) for(recount in 1:length(issueemails))
positionOfTimeStamp[recCount]<-pOfTimeStamp[recCount][1]
dateTimeStamp<-vector(mode = "character", length=length(positionOfGuid))
for(recCount in 1:length(issueemails))
dateTimeStamp[recCount]<-ifelse(positionOfTimeStamp[recCount] > 0,
substring(issueemails[recCount], positionOfTimeStamp[recCount], position Of TimeStamp[recCount]+18), "")
```

Now the below code extracts email content of respective created document ids based on Guid pattern and time stamp pattern.

```
emailText<-vector(mode = "character", length=length(positionOfGuid))
for(recCount in 1:length(issueemails))
emailText[recCount]<-ifelse(positionOfTimeStamp[recCount] > 0,
substring(issueemails[recCount],
positionOfTimeStamp[recCount]+19,
nchar(issueemailsrecCount])-positionOfTimeStamp[recCount]),
substring(issueemails[recCount],
positionOfGuid[recCount]+36,
nchar(issueemails[recCount])-positionOfGuid[recCount]+35))
countguidsfull<-0
countguidspartial<-0
countguidsnull<-0
for(recCount in 1:length(GUID))
{
if(nchar(dateTimeStamp[recCount])<= 1 )
countguidsnull<-countguidsnull +1
if(nchar(dateTimeStamp[recCount]) > 1 & (nchar(GUID[recCount]))<19 )
countguidspartial<-countguidspartial +1
if(nchar(dateTimeStamp[recCount]) == 19 ) countguidsfull<-countguidsfull +1
}
countguidsfull<-0
countguidspartial<-0
countguidsnull<-0
for(recCount in 1:length(GUID))
{
if(nchar(GUID[recCount])<= 1 ) countguidsnull<-countguidsnull +1
if(nchar(GUID[recCount]) > 1 & (nchar(GUID[recCount]))<36 )
countguidspartial<-countguidspartial +1
if(nchar(GUID[recCount]) == 36 )
countguidsfull<-countguidsfull +1
}
```

By the above code extraction of emails based on Guid pattern and time stamp pattern are extracted and we will size down the documents set to 1000 to be able to run with the memory limit and code is as follows:

```
emailText<-str_trim(emailText)
emailText<-str_replace_all(emailText, "http", " http")
```

Initially connection to be established with the respective files and find details of records can be coded in R as shown in Appendix 1.

Since the extraction of all mails with the content based on guide and time stamp pattern is done which resulted in creation of individual text files.

Preprocessing of text: Text collected from websites will be in Indigenous format and cannot be analyzed. To analyze sentiments and then come to a conclusion through the text data of emails, the data collected must be in the correct format but data collected might contain many unimportant repeated or general punctuation words (like and or the) or might be formatted inconveniently, which also can be termed as Unstructured data[8]. Hence, there are thousands and millions of text data available on the web, especially in social media sites which interpreted properly can help to get valuable conclusions. But they are not in a correct format or not in a structured way to get maximum usage out of them[9]. Therefore, to perform further analysis of this type of data or unstructured data methods for cleaning up is done for removing all the unnecessary content[9].

Before performing preprocessing of text, we need to construct "Corpus" of all email text Documents using "tm" package in R. This is because we have a large collection of documents performing all the preprocessing techniques for each text document is time-consuming process which can be performed easily by constructing corpus In R corpus is created using the following code:

```
install.packages("tm")
docs<-Corpus(DirSource("path to your folder"))
example: docs<-Corpus(DirSource("D:/work"))
```

where, folder "Work" is a collection of all email text documents, present in "D" drive.

Now conversion of unstructured data into structured using preprocessing includes the following steps:

- Tokenization
- Lower case conversion
- Removal of numeric values
- Lemmatization also denoted as stemming
- Removal of stop words

- **Tokenization:** In this step, the whole content of the text is split into separate entities like words, symbol, punctuation marks, expressions or other important components called tokens. These tokens can be used for further mining of content
- **Lower case conversion:** In this step, the whole text needs to be converted into lower case
- **Removal of numeric values:** This includes removal of numbers i.e, numeric values from the whole content of the text
- **Lemmatization (stemming):** There may be derivationally related words with similar meanings in the text. This step includes removal of those derivational affixes to bring the word to its root form
- **Removal of stop words:** Every data base contains some commonly used words, those words need to be removed which can reduce text content thereby we can concentrate on keywords

All the above mentioned, preprocessing steps of test data obtained can be done by using the tm package present in R software And the R code related to this preprocessing techniques is as given below:

```
docs<-tm_map(docs, content_transformer(tolower))
docs<-tm_map(docs, removePunctuation)
docs<-tm_map(docs, stripWhitespace)
docs<-tm_map(docs, removeWords, stopwords("english"))
docs<-tm_map(docs, removeNumbers)
docs<-tm_map(docs, stemDocument)
```

Feature selection: Recent studies on email sentiment analysis reveals that there is no appropriate feature selection method to perform analysis of email data, only based on the identification of features of individual words present in the data helps us in selecting identifying the related feature selection expand methods like BoWs, term presence, DTM and TF-IDF model, these feature selection methods performance also based on length of data, document length and other factors related to individual words[6,10]. Our interest in the present paper related to feature selection is DTM (document-term matrix), which calculates term frequency of each term among all documents creating a numeric structure of documents fairly simple representation of all documents. The DTM can become very large, sparse matrix depending on the number of documents in the corpus and number of terms in each document wherein matrix row corresponds to documents in the corpus and columns corresponds to terms, this easy numerical representation helps in text mining analytics for ranking, for identification of similarity of documents and categorization and for application of several other machine learning algorithms.

```
dtm<-DocumentTermMatrix(docs)
```

When a text is considered, for achieving the aspect of deriving a word and its pertaining emotion or to conclude the text which signifies either positivity or negativity, analyzing further is required to attain the goal. The text mining tools are programmatically analyze the text and extract related emotions of the text data. To analyze the sentiment of a text we will regard the text as a sequence of all its individual words and the value of sentiment content of individual words denotes the sentiment content of the whole text, this is most rarely used method to analyze the sentiment. This programmatic evaluation is implemented using R and most popularly tidy tool ecosystem is used for performing analysis. Data principles and functions to perform text mining tasks easier are present in this package, most of the related principles of text mining are already exists in packages like dplyr, tidyr, ggplot 2, stringr, so along with tidy text package, it is necessary to include all these packages.

```
install.packages("stringr")
install.packages('tidytext')
install.packages('tidyr')
install.packages('ggplot2')
install.packages('dplyr')
td<-tidy(dtm)
```

Sentiment classification: The principle objective of this study was to detect the sentiments from email and to find the keywords from the email data set which determine the positive and negative attitude. There are various methods and lexicons available in the tidy text package in the sentiments dataset for identifying the emotion from text, all available lexicons are used, also these lexicons consider only single words i.e., unigrams as a base. It consists of many English words which are assigned with positive or negative sentiment and also possibly emotions like joy, anger, sadness and so forth. Hence, the three lexicons available are:

- **AFINN from Finn Arup Nielsen:** This assigns words with a score that runs between -5 and 5 where positive score indicates positive sentiment and a negative score indicating negative sentiment.
- **Bing from Bing Liu and collaborators:** This lexicon categories words into positive categories and negative categories
- **NRC from Saif Mohammed and Peter Turney:** The NRC lexicon Categorizes words in a binary manner as "yes or no" into categories of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust

All this information is contained in the sentiments dataset and function get_sentiments () present in tidy text package helps to know sentiment lexicons.

In the present context, consider "bing" lexicon using inner join as approach to categorize words into positive and negative.

RESULTS

As discussed in the previous section the feature selection method which we choose DTM is converted into a tidy format, which helps in finding each words sentiment score using the BING lexicon as explained earlier and inner join() function and the related code is as below.

```
p_sentiments<-td%>%inner_join(get_sentiments("bing"), by = c(term = "word"))
```

This p_sentiments obtained helps to get the attitudes and opinions of words expressed in the collection of documents and the contribution of each word to each sentiment. To visualize all the top positive and negative words present in the email text documents R syntax is :

```
p_sentiments%>%
count(sentiment, term, wt = count) %>%
ungroup()%>%
filter(n >= 20)%>%
mutate(n = ifelse(sentiment == "negative", -n, n))%>%
mutate(term = reorder(term, n))%>%
ggplot(aes(term, n, fill = sentiment))+
geom_bar(stat = "identity")+
ylab("Contribution to sentiment")+
coord_flip()
```

The resultant graph obtained of reviews using ggplot2 visualization is as below.

Figure 2 displays the words that intent positivity and negativity. Figure 2 can also be simplified for a better understand of the positive and negative words are also possible by the following syntax:

```
bing_word_counts<-td%>%
inner_join(get_sentiments("bing"), by = c(term = "word"))%>%
count(term,sentiment, sort = TRUE)%>%
ungroup()
bing_word_counts%>%
group_by(sentiment)%>%
top_n(10)%>%
ungroup()%>%
mutate(term = reorder(term, n))%>%
ggplot(aes(term, n, fill = sentiment))+
geom_col(show.legend = FALSE)+
facet_wrap(~sentiment, scales = "free_y")+
labs(y = "Contribution to sentiment", x = NULL) +coord_flip()
```

The resultant graph obtained of individual positive and negative words using ggplot2 visualization is as below.

Figure 3 is easily understandable and clearly extracts the words that confirm to the positivity opinion and negativity.

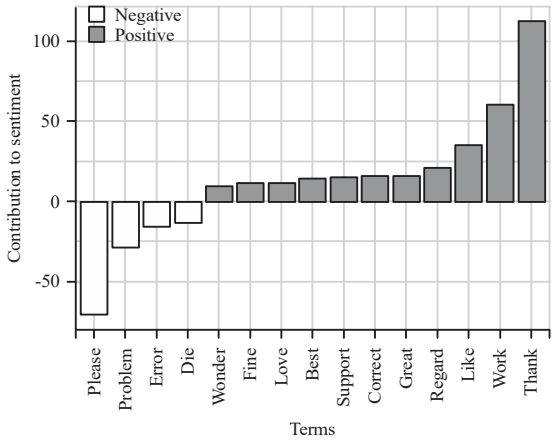

Fig. 2: Contribution to sentiment from email data

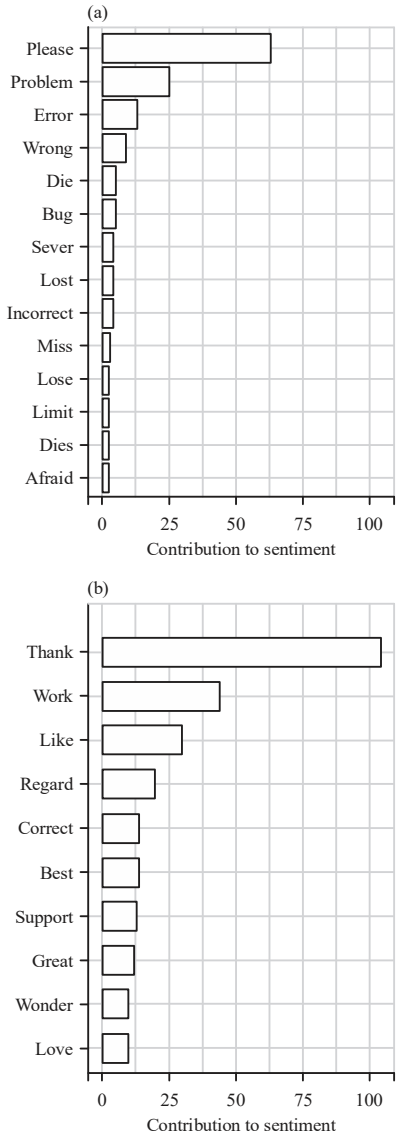

Fig. 3(a-b): Words that contribute to (a) Negative and (b) Positive sentiments of email data

DISCUSSION

From previous results Fig. 2 show that words like wonder, fine, love, best, support, correct, great, regard, like work, thank relates to positivity opinion and words like please, problem, error, die shows negative gratitude. And also Fig. 3 shows individually words that contribute to positive and negative opinion. Hence, identifying the words count of both sentiments and concentrating on the mails with words that confine the negative gratitude helps organizations to take further decisions to meet customer demands. Since it is not necessary that a sentiment analysis system should analyze each sentence or document content, it also needs to summarize some content of it to identify the nature of opinion like positive and negative, owing to the special characteristics of sentiment analysis.

As in recent studies in literature, sentiment analysis mainly focused on large scale social media microblogging sites like Facebook data, twitter corpus and also blog sites but not on email data as for email data analysis, previous research focused mainly on spam detection, social network filtering, management and priority issues[11-13] but there is no approach to perform sentiment analysis on email data, to identify the words that intent positive or negative opinion. Even some approaches measured polarity on the sentence flow of reviews using sequential conditional random fields[14] which is one of the approaches renowned as a cascaded approach which provides better results of polarity, but they did not consider lexicon based approach which provides fast and accurate results instance, tracking sentiment words visualization of archived emails is explored. But visualization of different sentiment words from emails which is not illustrated in recent approach is done in this study. Also experiments on the gender difference of email data sets based on sentiment axis is proposed[15]. However, which confines that less research is done for email sentiment analysis, the framework designed in this study and results obtained are in close agreement with the previous studies.

CONCLUSION

In the present the world is becoming narrower and with vast usage of internet can get the voice of people for particular products, events, issues very fast on web hence identifying the opinion from a large email database to know the voice of people is very important in analyzing the sentiments accordingly, it is also very important to analyze how people think in different context about different things. This becomes more important when it comes to the business world where we wants to estimate the current and future trends could be achieved by sentiment analysis. Also, using the product reviews and people opinions for a social issues and analyzing them helps to fulfill the customer requirements. In this article, analyzing the result of the email database and two classification tasks like positive and negative using a corpus-based approach is proposed. To improve efficiency and effectiveness of sentiment analysis in future we are concentrating on both objective sentences and subjective sentences.

In future work, we are also trying to consider newspaper articles, twitter data bases for classification, as well as thoroughly examine the impact on more sophisticated machine learning algorithms on classification accuracy, scrapping.

SIGNIFICANT STATEMENT

This study discovers the hidden emotional intents from emails to know the voice of people about a particular product which helps an organizations in estimating the current and future trends. This study will help the researchers to uncover the critical areas of a large email database that many researchers were not able to explore, Thus a new theory on email sentiment analysis using corpus based approach may be arrived at.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial and fellowship support granted by the Department of Science and Technology (DST-INSPIRE (Innovation in Science Pursuit for Inspired Research)),New Delhi, India. For carrying out the study. The authors also wish to thank Dr. S.A JyothiRani from Department of Statistics, Osmania University and Mokalla Thirupathi Reddy from National Institute of Nutrition (NIN) for encouragement and support.

REFERENCES

1. Chiu, C.M., 2004. Towards a hypermedia-enabled and web-based data analysis framework. J. Inform. Sci., 30: 60-72.
2. Aue, A. and M. Gamon, 2005. Customizing sentiment classifiers to new domains: A case study. Proceedings of the International Conference on Recent Advances in Natural Language Processing, September 21-23, 2005, Borovets, Bulgaria, pp: 207-218.
3. Zhang, L. and B. Liu, 2016. Sentiment Analysis and Opinion Mining. In: Encyclopedia of Machine Learning and Data Mining, Sammut, C. and G. Webb (Eds.)., Springer, Boston, MA.
4. Kolkur, S., G. Dantal and R. Mahe, 2015. Study of different levels for sentiment analysis. Int. J. Curr. Eng. Technol., 5: 768-770.
5. Liu, B., 2012. Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol., 5: 1-167.
6. Pang, B. and L. Lee, 2008. Opinion mining and sentiment analysis. Found. Trends Inform. Retrieval, 2: 1-135.
7. Mohammad, S.M., 2017. Challenges in Sentiment Analysis. In: A Practical Guide to Sentiment Analysis. Socio-Affective Computing, Vol. 5, Cambria, E., D. Das S. Bandyopadhyay and A. Feraco (Eds.)., Springer, Cham, ISBN: 978-3-319-55392-4, pp: 61-83.
8. Bafna, P., D. Pramod and A. Vaidya, 2016. Document clustering: TF-IDF approach. Proceedings of the 2016 International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT), Chennai, India, March 3-5, 2016, IEEE., pp: 61-66.
9. Vijayarani, S., J. Ilamathi and Nithya, 2015. Preprocessing techniques for text mining-an overview. Int. J. Comput. Sci. Commun. Netw., 5: 7-16.
10. Medhat, W., A. Hassan and H. Korashy, 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams Eng. J., 5: 1093-1113.
11. Hangal, S., M.S. Lam and J. Heer, 2011. Muse: Reviving memories using email archives. Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, October, 2011, ACM., New York, pp: 75-84.
12. Klimt, B. and Y. Yang, 2004. The enron corpus: A new dataset for email classification research. Proceedings of the European Conference on Machine Learning ECML, September 20-24, 2004, Springer, Pisa, Italy, pp: 217-226.
13. Whittaker, S. and C. Sidner, 1996. Email overload: Exploring personal information management of email. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground, Apr. 13-18, Vancouver, British Columbia, Canada, pp: 276-283.
14. Mao, Y. and G. Lebanon, 2006. Isotonic Conditional Random Fields and Local Sentiment Flow. In: Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, Schölkopf, B., J. Platt and T. Hofmann (Eds.)., MIT Press, UK., ISBN: 9780262256919, pp: 961-968.
15. Ravi, K. and V. Ravi, 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowl. Based Syst., 89: 14-46.